

Sole Power Mobile App - Technical Scope of Work

I. Functional Hierarchy

****1. Participant Role****

*** 1.1. Account Management**

*** 1.1.1. Sign Up**

*** 1.1.1.1. User Input**

*** 1.1.1.2. Validation**

*** 1.1.1.3. Account Creation**

*** 1.1.1.4. Welcome Message**

*** 1.1.2. Login**

*** 1.1.2.1. User Input**

*** 1.1.2.2. Authentication**

*** 1.1.2.3. Session Management**

*** 1.1.2.4. Access Control**

*** 1.1.3. Logout**

*** 1.1.3.1. Session Invalidation**

*** 1.1.3.2. Redirect**

*** 1.1.4. Profile Management**

*** 1.1.4.1. View Profile**

*** 1.1.4.2. Edit Profile**

*** 1.1.4.3. Password Reset**

*** 1.1.4.4. Email Address Change**

*** 1.2. Trip Tracking & Logging**

*** 1.2.1. Trip Tracking (GPS)**

*** 1.2.1.1. Start Trip**

*** 1.2.1.2. Trip Type Selection**

- * 1.2.1.3. Mode Selection
- * 1.2.1.4. Distance Calculation
- * 1.2.1.5. Trip End
- * 1.2.1.6. Auto-Populate Trip Log
- * 1.2.2. Manual Trip Entry
 - * 1.2.2.1. Trip Details Input
 - * 1.2.2.2. Validation
 - * 1.2.2.3. Trip Log Update
- * 1.3. Trip Log & Dashboard
 - * 1.3.1. Trip Log
 - * 1.3.1.1. View Trip Log
 - * 1.3.1.2. Filter Trips
 - * 1.3.1.3. Edit Trip
 - * 1.3.1.4. Export Trip Data
 - * 1.3.2. Personal Dashboard
 - * 1.3.2.1. Current Season Data
 - * 1.3.2.2. Calendar Year Data
 - * 1.3.2.3. Lifetime Data
 - * 1.3.2.4. Progress Visualization
- * 1.4. Team Management
 - * 1.4.1. Team Selection
 - * 1.4.1.1. View Available Teams
 - * 1.4.1.2. Join Team
 - * 1.4.1.3. Create Team
 - * 1.4.2. Team Leaderboards
 - * 1.4.2.1. View Team Leaderboard
 - * 1.4.2.2. Team Member Details
 - * 1.4.2.3. Historical Team Leaderboards
- * 1.5. Program Information & Engagement

- * 1.5.1. Program Overview
 - * 1.5.1.1. Program Description
- * 1.5.2. Challenges & Rewards
 - * 1.5.2.1. Challenge & Reward Display
- * 1.5.3. Resources & Information
 - * 1.5.3.1. Resource Links
- * 1.5.4. Social Sharing
 - * 1.5.4.1. Social Sharing Features
- * 1.5.5. Push Notifications
 - * 1.5.5.1. Push Notification System

****2. Administrator Role****

- * 2.1. User Management
 - * 2.1.1. Team Management
 - * 2.1.1.1. Team Requests
 - * 2.1.1.2. Team Member Management
 - * 2.1.2. User Account Management
 - * 2.1.2.1. Password Reset
 - * 2.1.2.2. Account Deactivation
 - * 2.1.2.3. User Data Export
- * 2.2. Trip Management
 - * 2.2.1. Trip Editing
 - * 2.2.1.1. Trip Editing Functionality
 - * 2.2.2. Trip Data Analysis
 - * 2.2.2.1. Trip Data Analysis Tools
- * 2.3. Program Management
 - * 2.3.1. Dashboard Management
 - * 2.3.1.1. View Real-Time Dashboard

- * 2.3.1.2. Reset Dashboard
- * 2.3.2. Newsletter Management
 - * 2.3.2.1. Create & Send Newsletters
 - * 2.3.2.2. Manage Subscriber List
- * 2.4. System Administration
 - * 2.4.1. Backend Access
 - * 2.4.1.1. Backend Access Control
 - * 2.4.2. System Monitoring
 - * 2.4.2.1. System Monitoring Tools
 - * 2.4.3. Security Management
 - * 2.4.3.1. Security Measures
- * 2.5. Future Program Expansion
 - * 2.5.1. New Program Features
 - * 2.5.1.1. Modular Architecture
 - * 2.5.2. Program Updates
 - * 2.5.2.1. Program Update Mechanism

II. Functional Details

****1. Participant Role****

*** **1.1. Account Management****

*** **1.1.1. Sign Up****

*** **Summary: Sign Up****

This task involves implementing a user-friendly sign-up form, validating user input, creating a new account with a unique ID, and displaying a welcome message.

* **1.1.1.1. User Input:** The sign-up form will be designed with a user-friendly interface, providing clear instructions and labels for each input field. The form will include fields for the user's name (first and last), email address, and password. The email address field will be validated using a regular expression to ensure a valid email format. The password field will enforce a minimum length

of 8 characters, require at least one uppercase letter, one lowercase letter, and one number, and will include a confirmation field to ensure the password is entered correctly. Optional fields will be provided for the user's team preference, age, and location. These fields will be optional to allow users to provide additional information if they choose.

****1.1.1.2. Validation:**** Server-side validation will be implemented to ensure the integrity and format of all user input data. Email validation will check for a valid email format and ensure that the email address is not already associated with an existing account. Password validation will enforce the minimum length, character complexity, and confirmation requirements. Age validation will ensure that the entered age is within a reasonable range.

****1.1.1.3. Account Creation:**** Upon successful validation, a unique user ID (UUID) will be generated for the new account. This ID will be used to identify the user within the system. User data will be stored securely in a database using appropriate encryption and hashing techniques. The password will be stored using a secure hashing algorithm like bcrypt, which prevents the storage of plain-text passwords.

****1.1.1.4. Welcome Message:**** After successful account creation, a welcome message will be displayed to the user. This message will provide a brief overview of the app's features and guide the user to the next steps, such as logging in or joining a team. The message will be designed to be informative and engaging, encouraging the user to explore the app's functionality.

****1.1.2. Login****

****Summary: Login****

This task involves implementing a secure login form, authenticating user credentials, managing user sessions, and controlling access based on user roles.

****1.1.2.1. User Input:**** The login form will be designed with a simple and intuitive interface, providing fields for the user's email address and password. A "Forgot Password" link will be provided below the form to allow users to reset their forgotten passwords.

****1.1.2.2. Authentication:**** Secure authentication will be implemented using a combination of email and password verification against the stored user data. The system will verify that the entered email address and password match the information stored in the database. To prevent brute-force attacks, rate limiting will be implemented to limit the number of login attempts within a specific time frame.

****1.1.2.3. Session Management:**** Upon successful authentication, a secure session will be created for the logged-in user. This session will be managed on the server-side using secure cookies or tokens. The cookies or tokens will be used to track the user's session on the client-side. Session timeout and inactivity logout mechanisms will be implemented to automatically log out users after a period of inactivity or when the session expires.

****1.1.2.4. Access Control:**** After successful login, the user will be redirected to the appropriate dashboard based on their role (participant). Role-based access control will be implemented to restrict access to specific features and functionalities based on the user's role.

*** **1.1.3. Logout****

*** **Summary: Logout****

This task involves invalidating the user's session and redirecting them to the appropriate page.

*** **1.1.3.1. Session Invalidation:**** When a user logs out, the current user session will be destroyed on the server-side. Any associated cookies or tokens on the client-side will be invalidated to prevent unauthorized access.

*** **1.1.3.2. Redirect:**** After logging out, the user will be redirected to the login page or a designated landing page. This will ensure that the user is no longer logged in and cannot access restricted areas of the app.

*** **1.1.4. Profile Management****

*** **Summary: Profile Management****

This task involves allowing users to view, edit, and manage their profile information, including password reset and email address change.

*** **1.1.4.1. View Profile:**** The user's profile information will be displayed in a dedicated section of the app. This information will include the user's name, email address, and team (if applicable). Optional fields, such as age and location, will be displayed if the user has provided this information during registration. Users will also be able to view their historical data, including their logged trips, mileage, and other relevant information.

*** **1.1.4.2. Edit Profile:**** Users will be able to update their profile information using a dedicated form. This form will allow users to edit their name, email address, and other optional fields. Data validation and security measures will be implemented to ensure that the updated information is accurate and secure.

*** **1.1.4.3. Password Reset:**** Users will be able to reset their forgotten passwords using a password reset functionality. This functionality will allow users to request a password reset link via email. A secure link will be sent to the user's email address, allowing them to set a new password.

*** **1.1.4.4. Email Address Change:**** Users will be able to change their associated email address using a dedicated form. This form will require the user to enter their current email address and the new email address. Email validation and confirmation will be implemented to ensure that the new email address is valid and that the user has access to it.

*** **1.2. Trip Tracking & Logging****

*** **1.2.1. Trip Tracking (GPS)****

*** **Summary: Trip Tracking (GPS)****

This task involves implementing GPS tracking functionality, allowing users to record their trips using their device's location services.

*** **1.2.1.1. Start Trip:**** The app will provide a "Start Trip" button that initiates GPS tracking. When the user clicks this button, the app will request permission to access the device's location services. Once permission is granted, the app will begin recording GPS coordinates and timestamps.

*** **1.2.1.2. Trip Type Selection:**** The app will provide a dropdown list or radio buttons for users to select the trip type. The available options will include "Work," "Errand," "Social," and "Other." The "Other" option will allow users to enter a custom description for their trip type.

*** **1.2.1.3. Mode Selection:**** The app will provide a dropdown list or radio buttons for users to select the mode of transportation. The available options will include "Walk," "Bike," "Run," "E-bike," "Skate," and "Other." The "Other" option will allow users to enter a custom description for their mode of transportation.

*** **1.2.1.4. Distance Calculation:**** The app will implement a distance calculation algorithm using the recorded GPS coordinates. The distance traveled will be calculated using the Haversine formula, which accurately calculates distances on a sphere. The distance will be displayed in real-time on the screen, allowing users to track their progress.

*** **1.2.1.5. Trip End:**** The app will provide an "End Trip" button to stop GPS tracking. When the user clicks this button, the app will stop recording GPS coordinates and timestamps. The recorded trip data, including the distance, trip type, mode, timestamps, and GPS coordinates, will be saved to the database.

*** **1.2.1.6. Auto-Populate Trip Log:**** The tracked trip data will be automatically added to the user's trip log. The trip details, including the date, distance, trip type, and mode, will be displayed in the trip log. This will provide users with a comprehensive record of their trips.

*** **1.2.2. Manual Trip Entry****

*** **Summary: Manual Trip Entry****

This task involves implementing a form for users to manually enter trip details, validate the input, and update the user's trip log.

*** **1.2.2.1. Trip Details Input:**** The app will provide a form for users to manually enter trip details. This form will include fields for the date, distance, trip type, and mode. The date field will be implemented using a date picker, allowing users to easily select the date of their trip. The distance field will be a numeric input field, allowing users to enter the distance of their trip. The trip type and mode fields will be dropdown lists, providing users with a selection of common options. Optional fields will be provided for additional details, such as the start location and end location.

*** **1.2.2.2. Validation:**** Server-side validation will be implemented to ensure the integrity and format of all manually entered trip data. The date field will be validated to ensure that the entered date is in a valid format. The distance field will be validated to ensure that the entered distance is a numeric value. The trip type and mode fields will be validated to ensure that the selected options are valid.

*** **1.2.2.3. Trip Log Update:**** The manually entered trip data will be added to the user's trip log. The trip details will be displayed in the trip log, providing users with a complete record of their trips.

*** **1.3. Trip Log & Dashboard****

*** **1.3.1. Trip Log****

*** **Summary: Trip Log****

This task involves implementing a user-friendly trip log that allows users to view, filter, edit, and export their trip data.

*** **1.3.1.1. View Trip Log:**** The trip log will display a list of all logged trips in chronological order. Each trip entry will include the date, distance, trip type, and mode. Users will be able to sort the trip log by any of these fields, allowing them to easily find specific trips.

*** **1.3.1.2. Filter Trips:**** The trip log will provide filtering options to allow users to narrow down the list of trips. Users will be able to filter trips by date range, trip type, and mode. This will allow users to focus on specific types of trips or trips within a specific time frame.

*** **1.3.1.3. Edit Trip:**** Users will be able to edit or delete individual trips from their trip log. This will allow users to correct any errors in their trip data or remove trips that are no longer relevant. Data validation and security measures will be implemented to prevent unauthorized modifications to trip data.

*** **1.3.1.4. Export Trip Data:**** Users will be able to export their trip log data in a downloadable format. The available export formats will include CSV and PDF. This will allow users to easily share their trip data with others or use it for analysis in other applications. Data security measures will be implemented to protect user privacy when exporting trip data.

*** **1.3.2. Personal Dashboard****

*** **Summary: Personal Dashboard****

This task involves implementing a dashboard that displays cumulative data for the current season, calendar year, and lifetime of the account, and provides progress visualization.

*** **1.3.2.1. Current Season Data:**** The personal dashboard will display cumulative data for the current season, which runs from Memorial Day to Columbus Day. The dashboard will display the following data points:

- * Number of trips:** The total number of trips logged during the current season.

- * Total mileage:** The total distance traveled during the current season.

- * Gas saved (gallons):** The estimated amount of gas saved by choosing alternative transportation modes.

- * CO2e reduced (Lbs):** The estimated amount of carbon dioxide emissions reduced by choosing alternative transportation modes.

- * Money saved (\$):** The estimated amount of money saved by choosing alternative transportation modes.

*** **1.3.2.2. Calendar Year Data:**** The personal dashboard will allow users to switch to view cumulative data for the current calendar year. This will provide users with a broader perspective on their transportation habits throughout the year.

*** **1.3.2.3. Lifetime Data:**** The personal dashboard will allow users to view cumulative data for the lifetime of their account. This will provide users with a long-term view of their transportation habits and progress.

*** **1.3.2.4. Progress Visualization:**** The personal dashboard will implement progress bars or charts to visualize the user's progress towards personal goals or challenges. Users will be able to set personal goals for mileage, trips, or CO2e reduction. This will help users stay motivated and track their progress towards their goals.

*** **1.4. Team Management****

*** **1.4.1. Team Selection****

*** **Summary: Team Selection****

This task involves allowing users to view available teams, join a team, and request to create a new team.

*** **1.4.1.1. View Available Teams:**** The app will display a list of available teams with basic information, such as the team name and the number of members. Users will be able to search for specific teams using a search bar. This will allow users to easily find teams that align with their interests or goals.

*** **1.4.1.2. Join Team:**** The app will provide a "Join Team" button for users to join a team. Users will be able to select a team from the list of available teams. The app will ensure that users can only join teams that are open for new members.

*** **1.4.1.3. Create Team:**** The app will provide a "Create Team" button for users to request to create a new team. Users will be required to provide a team name and a brief description. The team creation request will be submitted to the administrator for approval. This will ensure that all teams are reviewed and approved before they are made available to other users.

*** **1.4.2. Team Leaderboards****

*** **Summary: Team Leaderboards****

This task involves displaying team leaderboards with cumulative data for the current season, allowing users to view individual team member data, and providing access to historical team leaderboards.

*** **1.4.2.1. View Team Leaderboard:**** The app will display a leaderboard for each team with cumulative data for the current season. The leaderboard will display the following data points:

- * Team name:** The name of the team.

- * Number of current members:** The number of active members in the team.

- * Miles logged: The total distance traveled by all team members during the current season.
- * Trips logged: The total number of trips logged by all team members during the current season.
- * CO2e reduced: The estimated amount of carbon dioxide emissions reduced by all team members during the current season.
- * Gas saved (gallons): The estimated amount of gas saved by all team members during the current season.

* **1.4.2.2. Team Member Details:** Users will be able to view individual team member data by clicking on their name. This will display the trips logged and individual totals for the season, including mileage, CO2e reduction, and other relevant metrics.

* **1.4.2.3. Historical Team Leaderboards:** The app will implement a mechanism to access historical team leaderboards from previous years. This will allow users to compare team performance over time and track their progress.

* **1.5. Program Information & Engagement**

* **1.5.1. Program Overview**

* **Summary: Program Overview**

This task involves providing a detailed description of the Sole Power program, including its goals, objectives, benefits, and how it works.

* **1.5.1.1. Program Description:** The app will provide a detailed description of the Sole Power program, including its goals, objectives, and benefits. The description will be written in clear and concise language and will be visually appealing, using graphics and images to enhance the user experience. The app will also explain how the program works, including the rules and regulations.

* **1.5.2. Challenges & Rewards:**

* **Summary: Challenges & Rewards**

This task involves displaying current challenges and rewards for participants, allowing them to track their progress towards challenges.

* **1.5.2.1. Challenge & Reward Display:** The app will display current challenges and rewards for participants. This will include descriptions of the challenges, details of the rewards, eligibility criteria, and timelines. The app will implement a mechanism for users to track their progress towards challenges, allowing them to see how close they are to achieving their goals.

* **1.5.3. Resources & Information:**

* **Summary: Resources & Information**

This task involves providing links to relevant resources, such as bike maps, walking paths, and public transit schedules, and links to local businesses that offer discounts or incentives for sustainable transportation.

*** **1.5.3.1. Resource Links:**** The app will provide links to relevant resources, such as bike maps, walking paths, and public transit schedules. The app will also provide links to local businesses that offer discounts or incentives for sustainable transportation. The links will be up-to-date and functional, ensuring that users have access to the most relevant information.

*** **1.5.4. Social Sharing:****

*** **Summary: Social Sharing****

This task involves implementing social sharing features, allowing users to share their progress and achievements on social media platforms.

*** **1.5.4.1. Social Sharing Features:**** The app will implement social sharing features, allowing users to share their progress and achievements on social media platforms, such as Facebook, Twitter, and Instagram. Users will be able to share specific data points, such as their mileage, CO2e reduction, and other relevant metrics. The social sharing features will be secure and comply with privacy regulations.

*** **1.5.5. Push Notifications:****

*** **Summary: Push Notifications****

This task involves implementing a push notification system to send notifications to users about program updates, new challenges, upcoming events, and reminders to log trips.

*** **1.5.5.1. Push Notification System:**** The app will implement a push notification system to send notifications to users about program updates, new challenges, upcoming events, and reminders to log trips. Users will be able to customize their notification settings to receive only the notifications that are relevant to them. The push notifications will be timely and relevant, ensuring that users are kept informed about the program and its activities.

****2. Administrator Role****

*** **2.1. User Management****

*** **2.1.1. Team Management****

*** **Summary: Team Management****

This task involves allowing administrators to review and approve or deny team creation requests, and manage team members.

*** **2.1.1.1. Team Requests:**** The app will provide a dashboard for administrators to review and approve or deny team creation requests. The dashboard will display team creation requests with details, such as the team name, description, and creator information. Administrators will be able to approve or deny requests using buttons or checkboxes. The app will send notifications to team creators about the status of their requests.

****2.1.1.2. Team Member Management:**** The app will provide a dashboard for administrators to manage team members. This dashboard will allow administrators to add new members to existing teams, remove members from teams, and change team assignments for participants. The app will implement appropriate access control to prevent unauthorized modifications to team membership.

****2.1.2. User Account Management****

****Summary: User Account Management****

This task involves allowing administrators to manage user accounts, including password reset, account deactivation, and user data export.

****2.1.2.1. Password Reset:**** The app will provide a mechanism for administrators to force password reset for users. Administrators will be able to search for users by email address or user ID. Once a user is found, the administrator can initiate a password reset process. A secure link will be sent to the user's email address, allowing them to set a new password. The app will implement appropriate access control to prevent unauthorized modifications to user accounts.

****2.1.2.2. Account Deactivation:**** The app will provide a mechanism for administrators to deactivate user accounts. Administrators will be able to search for users by email address or user ID. Once a user is found, the administrator can deactivate the user's account. The app will implement appropriate access control to prevent unauthorized modifications to user accounts.

****2.1.2.3. User Data Export:**** The app will provide a mechanism for administrators to export user data. Administrators will be able to select the data fields to export, such as name, email, trips, and mileage. They will also be able to choose the export format, such as CSV or PDF. The exported data can then be downloaded by the administrator. The app will implement appropriate access control to prevent unauthorized access to user data.

****2.2. Trip Management****

****2.2.1. Trip Editing:****

****Summary: Trip Editing****

This task involves allowing administrators to edit or delete individual participant trips.

****2.2.1.1. Trip Editing Functionality:**** The app will provide a dashboard for administrators to edit or delete individual participant trips. Administrators will be able to search for trips by date, user ID, or trip details. Once a trip is found, the administrator can edit the trip details, such as the date, distance, trip type, and mode. They can also delete the trip. The app will implement appropriate access control to prevent unauthorized modifications to trip data.

****2.2.2. Trip Data Analysis:****

****Summary: Trip Data Analysis****

This task involves implementing tools for administrators to analyze trip data, generate reports, and visualize data using charts and graphs.

*** **2.2.2.1. Trip Data Analysis Tools:**** The app will implement tools for administrators to analyze trip data. This will include the ability to generate reports on total mileage logged, number of trips, CO2e reduction, gas saved, and the most popular trip types and modes. The app will also provide the ability to visualize data using charts and graphs. The data can be exported for further analysis in other applications.

*** **2.3. Program Management****

*** **2.3.1. Dashboard Management****

*** **Summary: Dashboard Management****

This task involves providing administrators with a dashboard to monitor program progress and key metrics, and allowing them to reset the dashboard at the start of each new challenge season.

*** **2.3.1.1. View Real-Time Dashboard:**** The app will provide a dashboard for administrators to monitor program progress and key metrics. The dashboard will display cumulative miles logged, CO2e reduction, total number of trips logged, number of active participants, and number of teams. The data will be visualized using charts and graphs to provide a clear and concise overview of program performance.

*** **2.3.1.2. Reset Dashboard:**** The app will provide a mechanism for administrators to reset the dashboard at the start of each new challenge season. This will involve clearing all cumulative data for the previous season and resetting the leaderboard rankings. The reset process will be secure and will not affect user data.

*** **2.3.2. Newsletter Management****

*** **Summary: Newsletter Management****

This task involves providing administrators with a system to create and send newsletters, and manage the subscriber list.

*** **2.3.2.1. Create & Send Newsletters:**** The app will provide a newsletter management system for administrators. This system will allow administrators to create new newsletters with content, subject line, and sender information. They will also be able to schedule newsletters to be sent at specific times. The app will track newsletter open rates and click-through rates to provide insights into newsletter performance.

*** **2.3.2.2. Manage Subscriber List:**** The app will provide a mechanism for administrators to manage the newsletter subscriber list. This will include the ability to add new subscribers, remove subscribers, and update subscriber information.

*** **2.4. System Administration****

*** **2.4.1. Backend Access:****

*** **Summary: Backend Access****

This task involves providing administrators with secure access to the backend database to manage user data, program settings, and system configurations.

*****2.4.1.1. Backend Access Control:**** The app will provide administrators with secure access to the backend database. This will allow administrators to manage user data, program settings, and system configurations. The app will implement role-based access control to restrict access to specific data and functions, ensuring that only authorized administrators have access to sensitive information. All backend operations will be logged and monitored for security purposes.

*****2.4.2. System Monitoring:****

*****Summary: System Monitoring****

This task involves implementing system monitoring tools to track server performance, database usage, app usage metrics, and error logs.

*****2.4.2.1. System Monitoring Tools:**** The app will implement system monitoring tools to track server performance, database usage, app usage metrics, and error logs. This will allow administrators to identify potential issues or performance bottlenecks and take corrective action.

*****2.4.3. Security Management:****

*****Summary: Security Management****

This task involves implementing security measures to protect user data and system integrity.

*****2.4.3.1. Security Measures:**** The app will implement security measures to protect user data and system integrity. This will include secure authentication and authorization, data encryption and hashing, regular security audits and vulnerability scans, and secure coding practices.

*****2.5. Future Program Expansion****

*****2.5.1. New Program Features:****

*****Summary: New Program Features****

This task involves designing the app with a modular architecture to allow for easy addition of new features, such as bus ridership tracking, carpooling, and other sustainable transportation options.

*****2.5.1.1. Modular Architecture:**** The app will be designed with a modular architecture to allow for easy addition of new features. This will enable the app to support future program expansions, such as bus ridership tracking, carpooling, and other sustainable transportation options. New features will be integrated seamlessly with existing functionality, ensuring a consistent user experience.

*****2.5.2. Program Updates:****

*****Summary: Program Updates****

This task involves implementing a mechanism for administrators to update program rules, incentives, or goals.

*****2.5.2.1. Program Update Mechanism:**** The app will implement a mechanism for administrators to update program rules, incentives, or goals. This will involve updating program

settings in the backend database and updating the app's content and UI to reflect the changes. Updates will be rolled out smoothly and will not disrupt user experience.

This technical scope of work provides a detailed breakdown of the functionalities required for both participant and administrator roles within the Sole Power mobile app. It serves as a comprehensive guide for the development team to ensure that the app meets all the requirements outlined in the RFP and is built with security, scalability, and future expansion in mind.